

## CLAIMS

We claim:

1 1. A method for control and management of communication  
2 traffic, comprising the steps of:

3 expressing access rules as filters referencing system  
4 kernel data;

5 for outbound processing, determining source application  
6 indicia;

7 for inbound packet processing, executing a look-ahead  
8 function to determine target application indicia; and

9 responsive to said source or target application  
10 indicia, executing filter processing.

1 2. The method of claim 1, further comprising the steps of  
2 executing said determining and executing steps within a

kernel filtering function upon encountering a filter  
selector field referencing kernel data not included in said  
packet.

3. The method of claim 1, said filter processing including  
the steps of:

determining a task or thread identifier;

based on said task or thread identifier, determining a  
process or job identifier; and

based on said process or job identifier, determining  
job or process attributes for filter processing.

4. The method of claim 1, said filter processing including  
the steps of:

determining a user identifier; and

based on said user identifier, determining user  
attributes for filter processing.

1 5. The method of claim 3, further comprising the step of  
2 determining from said task identifier a work control block  
3 containing said process or job identifier.

1 6. The method of claim 1, further comprising the steps for  
2 inbound processing of:

3 passing an inbound packet to a sockets layer to  
4 identify said target application.

1 7. The method of claim 6, further comprising the step of  
2 marking said inbound packet as not deliverable before  
3 passing it to said sockets layer.

1 8. The method of claim 1, further comprising the steps of:  
2 delivering to said filters infrastructure access rules  
3 for defining security context.

1 9. The method of claim 8, said infrastructure including

2 logging, auditing, and filter rule load controls.

1 10. A method for control and management of aspects of  
2 communication traffic within filtering, comprising the steps  
3 of:

4 receiving IP packet data into a TCP/IP protocol stack  
5 executing within a system kernel

6 executing filtering code within said system kernel with  
7 respect to non-IP packet data accessed within said  
8 system kernel outside of said TCP/IP protocol stack.

1 11. The method of claim 10, said non-IP packet data  
2 including context data regarding said IP packet.

1 12. The method of claim 10, said non-IP packet data  
2 including data specific to a task generating said non-IP  
3 packet data.

1 13. The method of claim 10, said non-IP packet data

2 including data specific to a task that will receive said IP  
3 packet.

1 14. The method of claim 11, said context data including  
2 packet arrival interface indicia.

1 15. The method of claim 11, said context data including  
2 packet arrival time-of-day indicia.

1 16. The method of claim 10, further comprising the steps  
2 of:

3 establishing a tunnel between two IP address limiting  
4 traffic to applications bound to ports at each end of  
5 said tunnel;

6 said filtering code accessing filtering attributes  
7 further limiting traffic selectively to job indicia.

1 17. The method of claim 10, further comprising the steps  
2 of:

3 establishing a tunnel between two IP address limiting  
4 traffic to applications bound to ports at each end of  
5 said tunnel; and

6 said filtering code accessing filtering attributes  
7 further limiting traffic selectively to user  
8 identification indicia.

1 18. A method for centralizing system-wide communication  
2 management and control within filter rules, comprising the  
3 steps of:

4 providing filter statements syntax for accepting  
5 parameters in the form of a selector, each selector  
6 specifying selector field, operator, and a set of  
7 values; and

8 said selector referencing data that does not exist in  
9 IP packets.

1 19. The method of claim 18, said parameters selectively  
2 including userid, user profile, user class, user group, user  
3 group authority, user special authority, job name, process

4 name, job group, job class, job priority, other job or  
5 process attributes, and date & time.

1 20. The method of claim 18, said filters statements being  
2 provided within a user interface to said system.

1 21. The method of claim 18, further comprising the steps  
2 of:

3 establishing a tunnel between two IP address limiting  
4 traffic to applications bound to ports at each end of  
5 said tunnel;

6 said filtering code accessing filtering attributes  
7 further limiting traffic selectively to job indicia;  
8 and

9 operating said filtering code within a kernel filtering  
10 function upon encountering a filter selector field  
11 referencing kernel data not included in said traffic.

1 22. A method for traversing a portion only of a protocol

2 stack to disallow selective IP packet traffic, comprising  
3 the steps of:

4 receiving a packet in the kernel of the operating  
5 system of a first node from an application, said kernel  
6 including a filter processor;

7 for inbound packet processing to a first node from a  
8 second node, executing a look-ahead function in the  
9 system kernel of said first node to determining a  
10 target application;

11 for both said inbound packet processing, and for  
12 outbound packet processing from said first node to said  
13 second node, executing within said kernel the steps of

14 processing said packet by determining a task ID;

15 responsive to said task ID, determining a  
16 corresponding work control block;

17 determining a user ID, process or job identifier  
18 from said work control block;

19 from the user ID, process or job identifier



20 selectively determining attributes for said user  
21 process or job; and  
  
22 passing said attributes to said filter processor  
23 for managing and controlling communication  
24 traffic.

1 23. A method for expressing access rules as filters,  
2 comprising the steps of:

3 providing a filter statements syntax for accepting  
4 parameters in the form of a selector, each selector  
5 specifying selector field, operator, and a set of  
6 values; and

7 said selector referencing data that does not exist in  
8 IP packets for controlling access to an application.

1 24. A method for managing and controlling communication  
2 traffic by centralizing access rules in filters executing  
3 within and referencing data available in system kernels,  
4 comprising the steps for outbound packet processing from a  
5 first node to a second node of:

6 receiving said packet in the kernel of the operating  
7 system of said first node from an application or  
8 process at said first node;

9 processing said packet by determining a task ID;

10 responsive to said task ID, determining a corresponding  
11 work control block;

12 responsive to said work control block, determining a  
13 process or job identifier;

14 responsive to said process or job identifier,  
15 determining job or process attributes.

1 25. The method of claim 24, further comprising the steps  
2 for inbound packet processing from said second node to said  
3 first node of:

4 initially operating said kernel at said first node to  
5 determine a target application for said packet at said  
6 first node.

1 26. The method of claim 25, said initially operating step  
2 comprising executing a look-ahead function.

1 27. The method of claim 26, said look-ahead function  
2 including the steps of operating a filter function to  
3 request of a sockets layer the identity of an application to  
4 which said sockets layer would pass said packet.

1 28. The method of claim 27, further comprising the step of  
2 marking said packet as non-deliverable and thereafter  
3 passing said packet to said sockets layer to identify said  
4 application.

1 29. A method for managing and controlling communication  
2 traffic by centralizing the access rules, comprising the  
3 steps for outbound packet processing from a first node to a  
4 second node of:

5 receiving said packet in the kernel of the operating  
6 system of said first node from an application or  
7 process at said first node, said kernel including a  
8 filter processor;

9 processing said packet by determining a task ID;

10 responsive to said task ID, determining a corresponding

11 work control block;

12 determining a user ID control block from said work

13 control block;

14 from the user ID control block determining attributes

15 for said user; and

16 passing said attributes to said filter processor for

17 managing and controlling communication traffic.

1 30. The method of claim 29, further comprising the steps

2 for inbound packet processing from said second node to said

3 first node of:

4 initially operating said kernel at said first node to

5 determine a target application for said packet at said

6 first node.

1 31. The method of claim 30, said initially operating step

2 comprising executing a look-ahead function.

1 32. The method of claim 31, said look-ahead function  
2 including the steps of operating a filter function to  
3 request of a sockets layer the identity of an application to  
4 which said sockets layer would pass said packet.

1 33. The method of claim 32, further comprising the step of  
2 marking said packet as non-deliverable and thereafter  
3 passing said packet to said sockets layer to identify said  
4 application.

1 34. A method for control and management of communication  
2 traffic with respect to a system node, comprising the steps  
3 of:

4 receiving at said system node an inbound packet; and

5 executing within a protocol stack of the system kernel  
6 of said system node a filtering function identifying  
7 for said inbound packet a filter referencing non-packet  
8 data; and

9 responsive to said filter, executing a look-ahead  
10 function for identifying a target application for said  
11 inbound packet.

1 35. The look-ahead function of the method of claim 34  
2 further comprising the steps of:

3 passing to a transport layer function identified by an  
4 IP header a packet marked non-deliverable for  
5 determining which user-level process or job is to  
6 receive said packet;

7 receiving from said transport layer an application  
8 layer task identifier for said user-level process or  
9 job; and thereafter

10 passing said packet marked by said task identifier to  
11 said transport layer for delivery to said application  
12 layer task.

1 36. System for control and management of communication  
2 traffic, comprising:

3 a system kernel including a filter function and stack  
4 data;

5 said filter function including a filter selectively  
6 referencing said stack data for expressing access  
7 rules;

8 said filter function being responsive to receipt of an  
9 outbound packet for determining a source application;

10 said filter function being responsive to receipt of an  
11 inbound packet processing for executing a look-ahead  
12 function to determine a target application; and

13 said filter function being responsive to said source or  
14 target application for executing filter processing.

1 37. A system for control and management of aspects of  
2 communication traffic within filtering, comprising:

3 a system kernel;

4 a protocol stack executing within said system kernel  
5 for receiving IP packet data; and

6 filtering code within said system kernel operable with  
7 respect to non-IP packet data accessed within said  
8 system kernel outside of said protocol stack for  
9 controlling and managing said aspects of communication  
10 traffic.

1 38. A system for centralizing system-wide communication  
2 management and control within filter rules, comprising:

3 filter statements having a syntax for accepting  
4 parameters in the form of a selector, each selector  
5 specifying selector field, operator, and a set of  
6 values; and  
7  
8 said selector referencing data that does not exist in  
IP packets.

1 39. A system for traversing a portion only of a protocol  
2 stack to disallow selective IP packet traffic, comprising:

3 a system kernel;

4 a filter processor executing within said system kernel;



5 said filter processor responsive to an inbound packet  
6 for executing a look-ahead function for determining a  
7 target application;

8 said filter processor responsive to both inbound and  
9 outbound packets for

10 processing said packet by determining a task ID;

11 responsive to said task ID, determining a  
12 corresponding work control block;

13 determining a user ID, process or job identifier  
14 from said work control block;

15 from the user ID, process or job identifier  
16 selectively determining attributes for said user  
17 process or job; and

18 passing said attributes to said filter processor  
19 for managing and controlling communication  
20 traffic.

1 40. A system for expressing access rules as filters,

2 comprising:

3 a filter statements for accepting parameters in the  
4 form of a selector, each selector specifying selector  
5 field, operator, and a set of values; and

6 said selector referencing data that does not exist in  
7 IP packets for controlling access to an application.

1 41. A system for managing and controlling communication  
2 traffic by centralizing access rules in filters executing  
3 within and referencing data available in system kernels,  
4 comprising:

5 code for receiving a packet in the kernel of the  
6 operating system of a first node from an application or  
7 process at said first node;

8 code for processing said packet by determining a task  
9 ID;

10 code responsive to said task ID for determining a  
11 corresponding work control block;

12 code responsive to said work control block for  
13 determining a process or job identifier; and  
  
14 code responsive to said process or job identifier for  
15 determining job or process attributes.

1 42. A system for managing and controlling communication  
2 traffic by centralizing access rules, comprising:

3 a first system node;

4 a second system node;

5 a kernel of the operating system of said first system  
6 node including a kernel filter processor;

7 said kernel for receiving from an application or  
8 process at said first system node a packet for  
9 communication to said second system node;

10 said kernel further for processing said packet by  
11 determining a task ID; responsive to said task ID,  
12 determining a corresponding work control block;  
13 determining a user ID control block from said work

14 control block; from the user ID control block  
15 determining attributes for said user; and passing said  
16 attributes to said system kernel filter processor for  
17 managing and controlling communication traffic.

1 43. A system for control and management of communication  
2 traffic with respect to a system node, comprising:

3 a filtering function executing within a protocol stack  
4 of the system kernel of said system node identifying  
5 for an inbound packet a filter referencing non-packet  
6 data; and

7 a look-ahead function responsive to said filter for  
8 identifying a target application for said inbound  
9 packet.

1 44. A program storage device readable by a machine,  
2 tangibly embodying a program of instructions executable by a  
3 machine to perform method steps for control and management  
4 of communication traffic, said method steps comprising:

5 expressing access rules as filters referencing system

6 kernel data;  
7 for outbound processing, determining a source  
8 application;  
9 for inbound packet processing, executing a look-ahead  
10 function to determine a target application; and  
11 responsive to said source or target application,  
12 executing filter processing.

1 45. A program storage device readable by a machine,  
2 tangibly embodying a program of instructions executable by a  
3 machine to perform method steps for control and management  
4 of aspects of communication traffic within filtering, said  
5 method steps comprising:

6 receiving IP packet data into a TCP/IP protocol stack  
7 executing within a system kernel  
8 executing filtering code within said system kernel with  
9 respect to non-IP packet data accessed within said  
10 system kernel outside of said TCP/IP protocol stack.

1 46. A program storage device readable by a machine,  
2 tangibly embodying a program of instructions executable by a  
3 machine to perform method steps for centralizing system-wide  
4 communication management and control within filter rules,  
5 said method steps comprising:

6 providing filter statements syntax for accepting  
7 parameters in the form of a selector, each selector  
8 specifying selector field, operator, and a set of  
9 values; and

10 said selector referencing data that does not exist in  
11 IP packets.

1 47. A program storage device readable by a machine,  
2 tangibly embodying a program of instructions executable by a  
3 machine to perform method steps for managing and controlling  
4 communication traffic by centralizing access rules in  
5 filters executing within and referencing data available in  
6 system kernels, said method steps comprising:

7 receiving said packet in the kernel of the operating  
8 system of said first node from an application or  
9 process at said first node;

10 processing said packet by determining a task ID;  
11 responsive to said task ID, determining a corresponding  
12 work control block;  
13 responsive to said work control block, determining a  
14 process or job identifier;  
15 responsive to said process or job identifier,  
16 determining job or process attributes.

1 48. The program storage device of claim 47, said method  
2 steps further comprising for inbound packet processing from  
3 said second node to said first node:

4 initially operating said kernel at said first node to  
5 determine a target application for said packet at said  
6 first node.

1 49. A computer program product or computer program element  
2 for control and management of communication traffic  
3 according to the steps comprising:

expressing access rules as filters referencing system  
kernel data;

for outbound processing, determining a source  
application;

for inbound packet processing, executing a look-ahead  
function to determine a target application; and

responsive to said source or target application,  
executing filter processing.

50. A computer program product or computer program element  
for control and management of aspects of communication  
traffic within filtering according to steps comprising:

receiving IP packet data into a TCP/IP protocol stack  
executing within a system kernel

executing filtering code within said system kernel with  
respect to non-IP packet data accessed within said  
system kernel outside of said TCP/IP protocol stack.



1 51. A computer program product or computer program element  
2 for centralizing system-wide communication management and  
3 control within filter rules according to method steps  
4 comprising:

5 providing filter statements syntax for accepting  
6 parameters in the form of a selector, each selector  
7 specifying selector field, operator, and a set of  
8 values; and

9 said selector referencing data that does not exist in  
10 IP packets.

1 52. A computer program product or computer program element  
2 for managing and controlling communication traffic by  
3 centralizing access rules in filters executing within and  
4 referencing data available in system kernels according to  
5 method steps comprising:

6 receiving said packet in the kernel of the operating  
7 system of said first node from an application or  
8 process at said first node;

9 processing said packet by determining a task ID;

10 responsive to said task ID, determining a corresponding  
11 work control block;  
  
12 responsive to said work control block, determining a  
13 process or job identifier;  
  
14 responsive to said process or job identifier,  
15 determining job or process attributes.

1 53. The computer program product or element of claim 52,  
2 said method steps further comprising for inbound packet  
3 processing from said second node to said first node:

4 initially operating said kernel at said first node to  
5 determine a target application for said packet at said  
6 first node.